

UNITED STATES DISTRICT COURT  
NORTHERN DISTRICT OF CALIFORNIA  
SAN FRANCISCO DIVISION

ORACLE AMERICA, INC.

Plaintiff,

v.

GOOGLE, INC.

Defendant.

Case No. 3:10-cv-03561-WHA

**REBUTTAL EXPERT REPORT OF ANDREW HALL**

**CONTAINS HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY MATERIAL  
SUBJECT TO PROTECTIVE ORDER**

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

**TABLE OF CONTENTS**

<b>I. INTRODUCTION.....</b>	<b>1</b>
<b>A. Case Background .....</b>	<b>1</b>
<b>B. Materials Considered.....</b>	<b>2</b>
<b>II. SUMMARY OF OPINIONS.....</b>	<b>2</b>
<b>III. BACKGROUND .....</b>	<b>4</b>
<b>IV. DETAILED OPINIONS IN REBUTTAL TO DR. KEMERER’S REPORT .....</b>	<b>5</b>
<b>A. Sun/Oracle’s choice of the GPL-2.0-CE license for the OpenJDK class libraries         suggests that it believed that the benefits of the Classpath Exception outweighed         concerns about the lesser copyleft effects of the GPL-2.0-CE license in comparison         to the GPL-2.0 license.....</b>	<b>5</b>
<b>B. Dr. Kemerer fails to acknowledge the effect of the Classpath Exception in his report.         .....</b>	<b>9</b>
<b>C. Dr. Kemerer conflates the GPL-2.0 license with the GPL-2.0-CE license.....</b>	<b>13</b>
<b>D. Dr. Kemerer conflates the GPL-2.0-CE license with the LGPL license. ....</b>	<b>21</b>
<b>V. CONCLUSION .....</b>	<b>26</b>
<b>APPENDIX D – DOCUMENTS EXAMINED DURING REPORT PREPARATION.....</b>	<b>27</b>

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

**I. INTRODUCTION**

**A. Case Background**

1. I have been asked by Google to review and respond to paragraphs 160-182 of a redacted version of the Expert Report of Chris F. Kemerer, Ph.D., dated January 8, 2016. I understand the redactions were made by Oracle to remove information that it contends is confidential to Oracle.

2. My qualifications and my compensation are set forth in paragraphs 1, 5-12 and Appendix A of my Opening Report, which I incorporate herein by reference.

3. At this time, I have not created any demonstrative exhibits to be used as a summary of, or as support for, my opinions, setting aside the tables and graphics in this report (including tables and graphics incorporated by reference from my opening report). I reserve the right to create any additional summaries, tutorials, demonstrations, charts, drawings, tables, and/or animations that may be appropriate to supplement and demonstrate my opinions if I am asked to testify at trial.

4. I understand that discovery is still ongoing. I therefore reserve the right to supplement my opinions after I have had the opportunity to review deposition testimony or in light of additional documents that may be brought to my attention. I further understand that the Court has adopted an expert-discovery schedule that allows experts to file reply reports, depending on the burdens of proof. Notwithstanding that schedule, if Oracle or its experts change their opinions (either explicitly or implicitly) in such a manner as to affect my conclusions, I may supplement my opinions with any necessary reply or supplemental reports.

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

**B. Materials Considered**

5. In forming my opinions and preparing this rebuttal report, I have considered the materials cited and listed in this report, as well as the documents listed in Appendix D.<sup>1</sup>

6. I also have specialized knowledge of the matters set forth herein, and if called as a witness I could and would testify competently thereto.

7. My research and analysis of the materials, documents, allegations, and other facts in this case are ongoing, and if additional information becomes available through discovery and depositions in this action, I reserve the right to provide additional opinions.

**II. SUMMARY OF OPINIONS**

8. In my opinion, Sun/Oracle understood that licensing the OpenJDK class libraries under the GNU General Public License version 2 with the Classpath Exception (the “GPL-2.0-CE” license) meant that the OpenJDK class libraries would be subject to the terms of the Classpath Exception and that third parties could therefore distribute software linking with the OpenJDK class libraries without limiting the distributor’s choice of license for the linking software and without obligating the distributor to publish the corresponding source code for the linking software. In my opinion, Sun/Oracle must have concluded that the benefits of granting the exception and permitting such linked combinations with the OpenJDK class libraries exceeded any theoretical harm that could result from publishing the libraries under the weak-copyleft GPL-2.0-CE license rather than the strong-copyleft GNU General Public License version 2 (the “GPL-2.0” license). In my opinion, the decision to use the GPL-2.0-CE license rather than the GPL-2.0 license likely increased adoption and use of the Java programming

---

<sup>1</sup> I am starting the lettering of appendices to this rebuttal report at D in order to avoid having more than one Hall Appendix with the same letter designation.

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

language and OpenJDK, particularly among companies developing and distributing software under closed-source license terms.

9. In my opinion, had Google chosen to adopt GPL-2.0-CE licensed OpenJDK class libraries for use in Android, the adoption of Android by OEMs would not have been impeded. In my opinion, Dr. Kemerer has erroneously conflated the GPL-2.0 license with the GPL-2.0-CE license. The obligations imposed by the two licenses are substantively different, especially with respect to the OpenJDK class libraries. In my opinion, Dr. Kemerer’s conclusion in paragraph 182 of his report (“handset manufacturers in 2007-2010 would have resisted use and distribution of GPL licensed versions of Android, as it would limit the handset manufacturers’ ability to withhold as proprietary their own modifications to Android and would therefore create a business risk for them”) is wrong to the extent Dr. Kemerer seeks to apply it to a potential version of Android that would have used the GPL-2.0-CE licensed class libraries from OpenJDK. As applied to that set of circumstances, based upon Anwar Ghuloum’s testimony, the materials I cite this report, and my own specialized knowledge and experience, it is my opinion that OEMs would not have had significant or lasting concerns in 2007 (or later years) if the 37 Java API packages had been licensed under the GPL-2.0-CE license.

10. The obligations imposed by the GNU Lesser General Public License (the “LGPL” license) are materially different from—and more onerous and restrictive than—the obligations imposed by the GPL-2.0-CE license. In my opinion, Dr. Kemerer therefore erred in relying on evidence related to the LGPL license to draw conclusions about the GPL-2.0-CE license. The LGPL license examples discussed by Dr. Kemerer therefore do not change my opinion, based upon Anwar Ghuloum’s testimony, the materials I cite this report, and my own specialized knowledge and experience, that OEMs would not have had significant or lasting concerns in

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

2007 (or later years) with using the 37 Java API packages if they had been licensed under the GPL-2.0-CE license.

**III. BACKGROUND**

11. In paragraphs 28-43 of my opening report, I provided background on software packaging and open-source licensing. I also explained, in paragraph 34, what it means for an open-source license to be a “copyleft” license (sometimes also described as a “hereditary,” “reciprocal” or “viral” license). In paragraphs 35-43, I explained the difference between “strong-copyleft” open-source licenses (including the GPL-2.0 license), “weak-copyleft” open-source licenses (including the GPL-2.0-CE license), and “permissive” open-source licenses (including the Apache License version 2 (the “Apache-2.0” license)). I then provided more details about the Apache-2.0 license and various GNU licenses in paragraphs 44-69.

12. In paragraphs 70-80, I explained various business models related to open-source software. And in paragraphs 81-83, I explained how software platforms and stacks can be comprised of components that are licensed under different open-source licensing (or even closed-source licensing) terms.

13. In paragraphs 84-92, I offered background about software “linking.”

14. In paragraphs 102-110, I offered background about the architecture of Android, including the licenses applicable to various components of Android (including that the Linux kernel is licensed under the GPL-2.0 license with system exceptions, that drivers and software packages bundled with the Linux kernel for a particular device are often licensed under the GPL-2.0 license or another GPL-2.0-compatible open-source license, that Google-authored components are typically licensed under the Apache-2.0 license, and that a variety of open-source licenses apply to various other Android components).

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

15. Given their direct relevance to the opinions offered by Dr. Kemerer regarding licensing—and in particular the terms of the GPL-2.0 license with the associated Classpath Exception (that is, the GPL-2.0 and GPL-2.0-CE licenses)—I incorporate by reference paragraphs 28-92 and 102-110, including the related tables and graphics, from my opening report.

**IV. DETAILED OPINIONS IN REBUTTAL TO DR. KEMERER’S REPORT**

**A. Sun/Oracle’s choice of the GPL-2.0-CE license for the OpenJDK class libraries suggests that it believed that the benefits of the Classpath Exception outweighed concerns about the lesser copyleft effects of the GPL-2.0-CE license in comparison to the GPL-2.0 license.**

16. In paragraph 162 of his report, Dr. Kemerer opines that the open-source licenses chosen by Sun/Oracle for the OpenJDK class libraries “preserved their investment and potential revenue streams [by requiring] contribution back of code modifications or derivatives.” To the extent that Dr. Kemerer assumes that Sun/Oracle’s license choices were made with reasoned intent, I agree. However, Dr. Kemerer’s conclusion regarding the requirement to “contribute back”<sup>2</sup> modifications and derivatives of the OpenJDK class libraries ignores the fundamental characteristic of the selected GPL-2.0-CE license: that it does *not* require the “contribution back” of linked software. The GPL-2.0 license extends its copyleft requirements to distributed derivative works of GPL-2.0 software. The Free Software Foundation, publisher of the GPL-2.0 license, takes the position that linking with GPL-2.0 libraries creates a derivative work of those libraries requiring that distributed copies of the combined software also be provided under the

---

<sup>2</sup> The term “contribution back” is often used in reference to the commonly held misunderstanding that most copyleft licenses require modifications or derivatives of the licensed open-source software to be assigned or licensed back to the copyright holders for the open-source software. In fact most copyleft software licenses (including the GPL-2.0 license) require that *recipients* of the open-source software be provided with the source code corresponding to the open-source software or that such source code be made specifically or generally available, in either case under the terms of the same copyleft license. I believe that Dr. Kemerer’s reference to “contributing back” is a reference to the copyleft obligations of those licenses, not the commonly held misunderstanding. I hereinafter refer to the obligation to offer or provide corresponding source code as an obligation to “publish” the source code.

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

GPL-2.0 license. However, the Classpath Exception of the GPL-2.0-CE license selected by Sun/Oracle specifically *exempts* software *linked with* the OpenJDK class libraries from the copyleft requirements of the GPL-2.0 license. Accordingly, Sun/Oracle selecting the GPL-2.0-CE license *rather than* the far more commonly used GPL-2.0 license<sup>3</sup> for the OpenJDK class libraries indicates that it believed that the best means to “preserve their investment and potential revenue streams” was to permit software distributors to link their application and other software with the OpenJDK class libraries *without* imposing copyleft requirements on the linking software.

17. As Dr. Kemerer suggests in paragraph 161 of his report, at least some sophisticated software developers choose the license terms for their software with an eye toward maximizing return. However, as explained in my opening expert report, maximizing return doesn’t necessarily correlate with maximizing control. Indeed, as indicated by the many different companies publishing proprietary open-source software under permissive (*i.e., not* copyleft) open-source licenses, sometimes *giving up* control *maximizes* return. In my opinion, the more reasonable conclusion drawn from Sun/Oracle’s selection of the GPL-2.0-CE license for the OpenJDK class libraries is that Sun/Oracle believed the benefits of permitting linking software combinations without imposing copyleft obligations on the linking software outweighed any theoretical harms that could result from the additional permissions granted under the Classpath Exception of the GPL-2.0-CE license.

---

<sup>3</sup> Open-source service provider Black Duck Software, which provides open-source scanning and auditing services, currently estimates that 22% of open-source projects use the GPL-2.0 license while less than 1% of open-source projects use the GPL-2.0-CE license. See Black Duck’s “Top 20 Open Source Licenses” available at <https://www.blackducksoftware.com/resources/data/top-20-open-source-licenses>. Notably, Sun distributed and Oracle distributes several other open-source packages under the GPL-2.0 license, including the MySQL database software and a customized version of the Linux operating system.



**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

18. Specifically, Sun/Oracle likely understood that publishing the OpenJDK class libraries under the GPL-2.0 license *without* granting an additional linking exception would negatively impact adoption and use of OpenJDK by commercial software developers stemming, in significant part, from software distributors’ concerns that closed-source software using the OpenJDK class libraries might *also* need to be published in source code form under the GPL-2.0 license. Such concerns are cited in a February 13, 2007, CNET article<sup>4</sup> describing the pending release of Java ME and SE under the GPL-2.0 license:

The nature of the GPL is that additions to software available under the GPL must also use the license. So a developer who writes an application using the open-source Java software could be obliged by the GPL to also release that newly created application as open source.

Citing Sun’s executive VP of software, Rich Green, the article explained that, for Java SE, “Sun is employing the so-called ‘classpath exception,’ a license addition that allows the company to place limits on the software that the GPL covers.” Mr. Green was quoted: “In the case of Java SE (Java Standard Edition), we’re enhancing (the GPL) with the classpath exception. *So when you’re working on top or shipping applications with the (Java) libraries and virtual machine, you’re not affected by the Java license.*” (*Emphasis added.*)

19. Mr. Green’s explanation, as summarized and quoted in the CNET article, was consistent with Sun’s published FAQ addressing the release of Java under an open-source license and its selection of the GPL-2.0-CE license (the “Java FAQ”).<sup>5</sup> The FAQ described the GPL-2.0-CE license and its selection for the core Java libraries (*emphasis added*):

---

<sup>4</sup> <http://www.cnet.com/news/sun-picks-gpl-license-for-java-code/>.

<sup>5</sup> The Java FAQ site originally posted by Sun has been taken down and replaced since Sun’s acquisition by Oracle. The original Java FAQ is still published in the OpenJDK section of the IcedTea project and is published available at <http://icedtea.classpath.org/openjdk/java/faq.jsp.html>. A snapshot of the original Java FAQ site taken June 4, 2011, by Archive.org is available at:

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

Q: What license did you choose for the open-source JDK components?

A: GPL v2 for almost all of the virtual machine, *and GPL v2 + the Classpath exception for the class libraries and those parts of the virtual machine that expose public APIs.*

Q: What is the Classpath exception?

A: The Classpath exception was developed by the Free Software Foundation's GNU/Classpath Project (see <http://www.gnu.org/software/classpath/license.html>). *It allows you to link an application available under any license to a library that is part of software licensed under GPL v2, without that application being subject to the GPL's requirement to be itself offered to the public under the GPL.*

Q: Why did you choose this licensing method?

A: This is the licensing paradigm in common use within Free software communities such as GNU/Classpath and Kaffe for the components of a Java technology implementation including the virtual machine and class libraries. *We consciously chose the same licensing method so that there would be no temptation to second guess Sun's intention to make its Java SE implementation available under a genuinely Free and open license and to allow easy collaboration with these existing communities.*

20. Mr. Green’s explanation was also consistent with the description of the Classpath

Exception published in the GNU Foundation’s Classpath FAQ:<sup>6</sup>

What does the exception allow me to do?

If you combine GNU Classpath with independent modules to produce an executable you can copy and distribute the resulting executable under terms of your choice.

So you can use and distribute GNU Classpath as is in your program without changing the license of your software.

21. Based on my review of the documents cited above and on my expertise in the field, it is my opinion that Sun/Oracle understood that licensing the OpenJDK class libraries under the GPL-2.0-CE license meant “the class libraries and those parts of the virtual machine

---

<https://web.archive.org/web/20110604004915/http://www.sun.com/software/opensource/java/faq.jsp>.

Contemporaneous versions were also produced in this case as GOOG-00000221 and GOOG-00000316.

<sup>6</sup> [http://www.gnu.org/software/classpath/faq/faq.html#faq2\\_3](http://www.gnu.org/software/classpath/faq/faq.html#faq2_3) (emphasis in the original).

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

that expose public APIs” would be subject to the terms of the Classpath Exception and that third parties could distribute software linking with the OpenJDK class libraries without limiting the distributor’s choice of license for the linking software and without obligating the distributor to publish the corresponding source code for the linking software. In my opinion, Sun/Oracle must have concluded that the benefits of granting the exception and permitting such linked combinations with the OpenJDK class libraries exceeded any theoretical harm that could result from publishing the libraries under the weak-copyleft GPL-2.0-CE license rather than the strong-copyleft GPL-2.0 license. In my opinion, the decision to use the GPL-2.0-CE license rather than the GPL-2.0 license likely increased adoption and use of the Java programming language and OpenJDK, particularly among companies developing and distributing software under closed-source license terms.

**B. Dr. Kemerer fails to acknowledge the effect of the Classpath Exception in his report.**

22. In paragraph 163 of his report, Dr. Kemerer opines that “a platform that was highly controlled by Sun may have created risk for Google in terms of Google’s ability to move into the market quickly and to take development of the platform wherever Google wished.” He then ties this notion to copyleft licensing terms, saying that “a company in Google’s position would not be amenable to licensing terms (for example, ‘copyleft’ open source licenses) that would create risks for potential OEM customers who might have to relinquish control over changes that they made to the platform.”

23. First, Dr. Kemerer overstates the copyleft requirements of the GPL-2.0-CE license when he concludes that OEM’s “might have to relinquish control over changes that they made to the platform.” Given the manner in which the OpenJDK class libraries are being incorporated into the Android software stack, OEMs would only need to publish changes made

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

to the OpenJDK class libraries themselves; changes made elsewhere in the Android platform would not be subject to this requirement.

24. Accordingly, including the GPL-2.0-CE-licensed libraries rather than the accused Apache-2.0-licensed API libraries would impact only those OEMs that both (a) desire to modify the OpenJDK class libraries and (b) object to publishing those modifications in source code form under the GPL-2.0-CE license. If an OEM distributor doesn’t modify the OpenJDK class libraries or modifies the OpenJDK class libraries but doesn’t object to publishing those modifications under the GPL-2.0-CE license, then the copyleft requirements of the weak-copyleft GPL-2.0-CE license would not discourage the OEM from adopting and distributing the Android software stack.

25. Further, Dr. Kemerer’s conclusion fails to account for Android’s longstanding inclusion of many other components licensed under strong-copyleft or weak-copyleft licenses. The Linux kernel, which provides the base of the Android software stack, is licensed under the strong-copyleft GPL-2.0 license. Other portions of the Android platform are also licensed under multiple other copyleft licenses. Here are just a few examples:

Component/file names and links to source code	License
Portions of the elfutils component including, for example: <a href="#">platform_external_elfutils/libelf/elf64_getchdr.c</a> <a href="#">platform_external_elfutils/lib/xmalloc.c</a> <a href="#">platform_external_elfutils/libasm/asm_abort.c</a>	Either: * LGPL-3.0; or * GPL-2.0.
Elfutils libelf components including, for example: <a href="#">platform_external_elfutils/libelf/dl-hash.h</a>	LGPL-2.1
Portions of the webkit component including, for example: <a href="#">platform_external_webkit/Source/WebCore/rendering/ScrollBehavior.h</a>	Either: * LGPL-2.1; * MPL-1.1; or * GPL-2.0.
Portions of the webkit component including, for example: <a href="#">platform_external_webkit/Source/JavaScriptCore/runtime/BooleanPrototype.h</a> ; <a href="#">platform_external_webkit/Source/JavaScriptCore/runtime/NumberPrototype.h</a> ; <a href="#">platform_external_webkit/Source/WebCore/platform/efl/KURLEfl.cpp</a> ; and <a href="#">platform_external_webkit/Source/WebCore/platform/network/soup/GOwnPtrSoup.cpp</a>	LGPL-2.0

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

26. When considering whether to include software published under a copyleft license in combination with closed-source software, the primary concern of most commercial software distributors is whether the proposed use will obligate the distributor to make its competitively valuable or sensitive software available in source code form under a copyleft license. Unlike strong-copyleft licenses (such as the GPL-2.0 license), the copyleft requirements of weak-copyleft licenses typically extend only to *modifications* to the distributed weak-copyleft software. The GPL-2.0-CE license similarly limits its copyleft impact by *excluding* linked software from the GPL-2.0 license’s copyleft requirements.

27. Thus, in considering whether Google would have found it acceptable to license the 37 Java API packages under the GPL-2.0-CE license, an essential question is whether OEMs desire to make changes to those packages without having to license those changes under the GPL-2.0-CE license. When asked whether he believed OEMs would object if the 37 Java API packages in Android were licensed under the GPL-2.0-CE license, Anwar Ghuloum, the Google Director of Engineering responsible for these portions of the Android platform, testified, “I don’t think so” and explained, “It’s – the difficulty here is that it’s unlikely that any OEM partner would actually add libraries to the OpenJDK packages.”<sup>7</sup> Instead, the changes that OEMs made that he was aware of were to “the applications frameworks in Android.”<sup>8</sup>

28. In paragraphs 118-122 of my opening report, which I incorporate by reference, I considered the extent to which the GPL-2.0-CE license that is applicable to the OpenJDK class libraries would have a copyleft effect on other software in the Android platform, if OpenJDK class libraries were to be incorporated into the platform. In my opinion, there would be no such copyleft effect. Stated more specifically, I considered the effect of the Android Runtime loading,

---

<sup>7</sup> Ghuloum depo. at 36:17, 37:7-10.

<sup>8</sup> *Id.* at 37:16-19.

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

linking, and executing OpenJDK-based API packages by or with (1) Android applications, (2) the Native Libraries, (3) other parts of the Android Runtime, (4) other Android Core Libraries, (5) the Application Framework or (6) any other portion of the Android platform. As I explained in paragraphs 118-122 of my opening report, in my opinion, this would not result in any components other than the OpenJDK-based Java API packages becoming otherwise subject to the GPL-2.0 license, the GPL-2.0-CE license, or any other open-source license. Put another way, the use of OpenJDK-based API packages would not have a copyleft effect on other software in the Android platform. Instead, the GPL-2.0-CE license’s copyleft effects would be limited to those OpenJDK-based API packages themselves. Like other weak-copyleft licenses, the GPL-2.0-CE license would impose its copyleft requirements only on distributed modifications to those OpenJDK-based Java API packages, not to software linking with those packages.

29. Thus, unless OEMs were concerned about having to open-source changes they make to *the OpenJDK-based Java API packages themselves*, Google would have no reason to avoid the copyleft impact of adopting the GPL-2.0-CE license for those packages. And as Anwar Ghuloum testified, there is no reason to believe that OEMs would have such a concern.

30. Dr. Kemerer in paragraph 182 of his report notes that Mr. Ghuloum also testified that he could not say “with certainty” that OEMs would have no such concern, but he also testified that he didn’t think OEMs would have such a concern, that it was unlikely that OEMs would want to make changes to the Java API libraries, that he was unaware of OEMs having made changes to the core libraries, and that it was “highly unlikely” that OEMs would reject the use of GPL-2.0-CE code in the Java API packages.<sup>9</sup> He also testified, “We deal with GPL code with OEMs all the time. There are parts of the Android platform that are GPL’d and we have

---

<sup>9</sup> Ghuloum depo. at 36:12-37:19, 59:12-19.

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

GPL obligations as a result.”<sup>10</sup> He noted that the Linux kernel that is part of Android is licensed under a GPL license, and later testified that Webkit (which is also part of Android) might be licensed under a GPL license, as well.<sup>11</sup> Mr. Ghuloum’s recollection was correct—as I noted in paragraph 25, above, portions of Webkit are licensed under the LGPL license.

31. Dr. Kemerer offers no concrete basis to believe that OEMs would be concerned about the presence of GPL-2.0-CE licensed code in the Java API packages. While he does note, in footnote 85, that OEMs may want to make additions or changes to APIs *in the application frameworks or libraries other than the core libraries*, as I have already explained, the presence of GPL-2.0-CE code in the Java API packages would not impose a copyleft effect on code in the application frameworks or any libraries other than the core libraries. Moreover, given that OEMs already accept LGPL obligations with respect to portions of certain components (such as Webkit) when they distribute Android, it is my opinion that adopting the less restrictive GPL-2.0-CE license<sup>12</sup> for some or all of the Java API packages would not have negatively impacted the adoption of Android by OEMs.

**C. Dr. Kemerer conflates the GPL-2.0 license with the GPL-2.0-CE license.**

32. In his report, Dr. Kemerer repeatedly conflates the GPL-2.0 license with the GPL-2.0-CE license. It also appears that Dr. Kemerer may not understand the purpose or impact of the Classpath Exception. By failing to distinguish between these licenses, Dr. Kemerer reaches several conclusions about the *GPL-2.0-CE* license based on evidence about the *GPL-2.0* license. In my opinion, Dr. Kemerer’s opinions are incorrect and unsupported.

---

<sup>10</sup> *Id.* at 34:16-19.

<sup>11</sup> *Id.* at 34:20-23, 62:8-10.

<sup>12</sup> I discuss the differences between LGPL and the GPL-2.0-CE license in greater depth in section IV.D of this report.

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

33. In paragraph 164 of his report, Dr. Kemerer states:

I have reviewed some of the history of Sun’s and Google’s interactions regarding an “open source” GPL license, and public documents reflecting Google’s position about that license at the time that it was attempting to introduce Android into the market. It is my opinion that Google did not want take [sic] such a license because important potential device manufacturer partners likely would not risk accepting code encumbered by this type of license.

In paragraph 165 of his report, Dr. Kemerer goes on to use the defined term “GPL” to refer specifically to “the GPLv.2 license with the Classpath exception”—*i.e.*, the GPL-2.0-CE license. But in subsequent paragraphs, Dr. Kemerer repeatedly cites to discussions of “GPL” licenses that do not include the Classpath Exception (that is, not the GPL-2.0-CE license).

34. For example, in paragraph 166 of his report, Dr. Kemerer cites an August 11, 2007, email from Google employee Andy Rubin to Google employee Bob Lee.<sup>13</sup> In that email, Mr. Rubin states, “as far as GPL-ing the VM, everything that is linked with the VM would get infected.” From that sentence, it is clear that Mr. Rubin is talking about a variant of the GPL that does *not* include the Classpath Exception (or any other linking exception), because he says that “everything” that is “linked” with the VM would get “infected”—*i.e.*, would be subject to GPL obligations. The very purpose of a linking exception is to allow at least some code that is linked with the GPL-licensed code to be licensed under terms *other* than the GPL. Thus, I disagree with Dr. Kemerer to the extent he is claiming that the GPL license being discussed in this email is the GPL-2.0-CE license. It is also worth noting that Mr. Rubin was discussing linking with “the VM,” or virtual machine, not linking with the class libraries. As is noted in the Java FAQ, which I have already quoted above, most of the VM in OpenJDK is licensed under the GPL-2.0 license, rather than the GPL-2.0-CE license that applies to the OpenJDK class libraries.

---

<sup>13</sup> GOOGLE-02-00020474.



**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

35. In the same email, Mr. Rubin goes on to explain:

The problem with GPL in embedded systems is that it's viral, and there is no way (for example) OEMs or Carriers to differentiate by adding proprietary works. We are building a platform where the entire purpose is to let people differentiate on top of it.

Notably, for a work licensed under the GPL-2.0-CE license, distributors can “differentiate on top of it” with proprietary code by either (a) restricting combinations between the GPL-2.0-CE code and the differentiating proprietary code to linking *or* (b) combining the GPL-2.0-CE code with the distributor’s proprietary code without creating a derivative work of the GPL-2.0-CE code. This confirms that Mr. Rubin was not referring to the GPL-2.0-CE license when he referred to “GPL” in this email.

36. In this email, Mr. Rubin next says:

Finally, Sun has a different license for its library for SE and ME. The SE library is LGPL, ME library is GPL. That means anything that links with the ME library gets infected. And the SE library is not optimized for embedded systems.

Here again, Mr. Rubin refers to the issue of GPL code “infect[ing]” other code that “links with” it. Notably, the code he is talking about—the “ME library”—was distributed by Sun with the GPL-2.0 license *without* the Classpath Exception.

37. Indeed, the distinction between Sun’s open-source license for Java ME and its license for Java SE was well-known at the time. As I have already noted above, Sun’s executive VP for software Rich Green explained to the tech industry press that Java ME and SE would both be licensed on an open-source basis under version 2 of the GPL, but that for *Java SE*, the license would include the Classpath Exception. As he explained, this was so “you’re not affected by the Java license” when “you’re working on top of shipping applications” with the Java SE

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

libraries.<sup>14</sup> Similarly, Sun’s Java FAQ, which I also have already discussed in this report, made clear that portions of OpenJDK subject to the Classpath Exception—“the class libraries and those parts of the virtual machine that expose public APIs”—enable an application to link to those components “without that application being subject to the GPL’s requirement to be itself offered to the public under the GPL.” Both Mr. Green’s explanation and the Java FAQ were also consistent with the GNU Foundation’s description of the Classpath Exception, as I have already discussed. Thus, it was well-known in 2007 that Sun’s open-source license for Java *SE* allowed developers to work “on top” of the OpenJDK class libraries and to “ship[] applications” without being forced to license their code on GPL terms.

38. In his August 11, 2007, email, Mr. Rubin also showed that he was aware that Sun’s open-source version of Java SE was licensed under less restrictive terms than the GPL-2.0 license. Mr. Rubin referred to this license as “LGPL,” which was not quite correct (because Sun’s open-source version of Java SE is licensed under GPL-2.0-CE), but that suggests to me that he recognized that Java SE was licensed under different terms than the GPL-2.0 license, and that the terms applicable to OpenJDK included a linking exception (such as those included in the LGPL license or the GPL-2.0-CE license). Further, when he explained his concern with Java SE, he did not mention the “viral” effect of GPL. Instead, Mr. Rubin said that his concern was that “the SE library is not optimized for embedded systems”—*i.e.*, Mr. Rubin identified a technical concern, not a licensing concern.

39. In paragraph 167 of his report, Dr. Kemerer moves on to discuss a January 4, 2007, email from Google employee Dan Bornstein to an email list called “thing-eng@google.com”. Dr. Kemerer notes that this email mentions the “GPL” license—again, a term

---

<sup>14</sup> <http://www.cnet.com/news/sun-picks-gpl-license-for-java-code/>.

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

Dr. Kemerer has equated with the GPL-2.0-CE license. But Mr. Bornstein’s email was *not* discussing the GPL-2.0-CE license. Mr. Bornstein’s email was in reply to January 4, 2007, email from Carl Quinn, another Google employee. Mr. Bornstein’s email quotes Mr. Quinn as having said, “Any Open Source Java \_ME\_ has no classpath exception . . . .” Mr. Bornstein responds, “Without the CLASSPATH exception, any app that gets loaded into a GPLed ME build must itself be GPLed (or violate the license).” Clearly, then, Mr. Bornstein’s concern was not with the GPL-2.0-CE license—because GPL-2.0-CE *includes* the Classpath Exception, it is not “[w]ithout the CLASSPATH exception.” Rather, by expressly pointing out that the *lack* of the Classpath Exception triggered his GPL concern, one can only conclude that Mr. Bornstein understood that code licensed *with* the Classpath Exception did not have the same viral effect as the GPL-2.0 license. Thus, I disagree with Dr. Kemerer to the extent that he assumes that Mr. Bornstein was using “GPL” to refer to the GPL-2.0-CE license in that email.

40. Mr. Bornstein goes on to say the “GPL” license “makes it impossible to create ‘extended subsets’ of the libraries based on the GPLed version . . . .” The previous portion of the email is discussing “Java ME,” which Sun had licensed *without* the Classpath Exception. I can only conclude that when Mr. Bornstein referred to the inability to create “extended subsets” of the libraries, he was referring to Sun’s open-source version of Java ME—*i.e.*, to code licensed under the GPL-2.0 license, not the GPL-2.0-CE license.

41. Thus, when Dr. Kemerer in paragraph 168 concludes that Google would not take “the GPL licensed version of Java as the basis for Android”—and, again, Dr. Kemerer defined “GPL” to mean the GPL-2.0-CE license—that conclusion is untethered from anything in the Rubin and Bornstein emails cited as support for that conclusion, because neither of those emails

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

discusses concerns about licensing restrictions imposed on software linking with the code subject to the Classpath Exception (i.e., code provided subject the terms of the GPL-2.0-CE license).

42. After paragraph 168 of his report, Dr. Kemerer has an indented paragraph. The formatting of that paragraph would appear to indicate that it is a quotation, but the content of the paragraph leads me to believe that this formatting is a mistake, and that this paragraph is actually Dr. Kemerer’s wording. Assuming that to be the case, Dr. Kemerer says, “In this way, modifications or derivatives of the APIs in the core libraries created by phone OEMs would, themselves, be subject to the terms of the GPL license and could not be withheld by the phone OEMs as proprietary - product differentiation being part of OEM competition.” I disagree with several portions of Dr. Kemerer’s statement and, in particular, Dr. Kemerer’s apparent conclusion that a restriction on the ability of OEMs to change the Java API core libraries without the ability to keep those changes as closed-source software would present a significant concern for OEMs. Again, Mr. Ghuloum testified that he did not believe that OEMs would want to make such changes. Instead, he testified that he was aware of changes only *to the application frameworks*. Such changes would either not be subject to the copyleft requirements of either the GPL-2.0 license or the GPL-2.0-CE license or would be made to packages qualifying for the Classpath Exception. In the 2007 words of Sun VP Rich Green: “So when you’re working on top or shipping applications with the (Java) libraries and virtual machine, you’re not affected by the Java license.”<sup>15</sup>

43. Dr. Kemerer goes on to state, “If Google had taken the GPL licensed version of Java, but Google or its phone manufacturer customers had tried to hold back any modifications that they made to Android, I have been informed that they would face the risk of a copyright

---

<sup>15</sup> <http://www.cnet.com/news/sun-picks-gpl-license-for-java-code/>.

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

lawsuit, since that activity could cause the license to terminate.” I disagree with this conclusion, because had Google adopted only some or all of the GPL-2.0-CE licensed OpenJDK class libraries, OEMs could have kept proprietary modifications to anything other than the OpenJDK class libraries, and that would not have terminated the GPL-2.0-CE license. As Sun VP Rich Green explained in 2007, the Classpath Exception allows software distributors to build “on top of” source code provided under the terms of the GPL-2.0-CE license without obligation to publish the distributor’s software under a GPL license. Mr. Green was correct—code in one or more independent modules that merely link with the GPL-2.0-CE software will not itself need to be licensed under the GPL-2.0-CE license. This feature allows OEMs to develop new code in, for example, the application frameworks, or in other code outside the OpenJDK class libraries, without needing to publish the OEMs’ developed code under the same open-source license. Moreover, as I have noted above and as Mr. Ghuloum testified, portions of Android are already licensed under GPL licenses, with the Linux kernel licensed under the GPL-2.0 license and portions of other components licensed under LGPL.

44. Dr. Kemerer, in footnote 86 to his report, also discusses several articles about lawsuits alleging non-compliance with open-source licenses. The first article discusses a lawsuit alleging a failure to comply with the GPL-2.0 license.<sup>16</sup> The article makes no mention of the license having the Classpath Exception or any other linking exception. The second article<sup>17</sup> again refers to an alleged failure to comply with the GPL-2.0 license and again makes no mention of the license having the Classpath Exception or any other linking exception. The third article,<sup>18</sup> like the others, makes no mention of the license having the Classpath Exception or any other

---

<sup>16</sup> <https://opensource.com/law/14/7/lawsuit-threatens-break-new-ground-gpl-and-software-licensing-issues>.

<sup>17</sup> <http://www.pcworld.com/article/2893852/vmware-sued-for-alleged-gpl-license-infractions.html>.

<sup>18</sup> <http://torquemag.io/2013/03/busybox/>.

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

linking exception. The third article also includes a link to the complaint, which identifies the license as the GPL-2.0 license.<sup>19</sup> The license is also attached as an exhibit to the complaint, and it does *not* include the Classpath Exception.

45. In paragraphs 174-180 of his report, Dr. Kemerer discusses an article about why Google chose the Apache-2.0 license over GPL-2.0 license as the primary license for Android.<sup>20</sup> Notably, that article contrasts the Apache-2.0 license with GPL-2.0 license and makes no mention of the Classpath Exception.

46. Dr. Kemerer also discusses, at paragraph 181 of his report, an article that discusses OpenJDK.<sup>21</sup> But that article is about the use of a full implementation of OpenJDK, which would include, for example, the OpenJDK virtual machine—much of which is licensed under the GPL-2.0 license rather than the GPL-2.0-CE license.

47. In sum, Dr. Kemerer repeatedly relies on evidence related to the GPL-2.0 license to draw conclusions about the GPL-2.0-CE license. Because the GPL-2.0 and GPL-2.0-CE licenses have substantively different copyleft effects, I consider this a major flaw in Dr. Kemerer’s reasoning and/or an indicator that Dr. Kemerer does not understand the effect of the Classpath Exception.

48. In my opinion, Dr. Kemerer’s conclusion in paragraph 182 of his report (“handset manufacturers in 2007-2010 would have resisted use and distribution of GPL licensed versions of Android, as it would limit the handset manufacturers’ ability to withhold as proprietary their own modifications to Android and would therefore create a business risk for them”) is wrong to the extent Dr. Kemerer seeks to apply it to a potential version of Android that would have used

---

<sup>19</sup> <http://torquemag-hhvm.s3.amazonaws.com/uploads/2013/03/SFLC-BusyBox-lawsuit.pdf>.

<sup>20</sup> <http://arstechnica.com/uncategorized/2007/11/why-google-chose-the-apache-software-license-over-gplv2/>.

<sup>21</sup> <http://www.javaworld.com/article/2078666/mobile-java/open-source-java-for-android--don-t-bet-on-it.html>.

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

the GPL-2.0-CE licensed class libraries from OpenJDK. As applied to that set of circumstances, based upon Anwar Ghuloum’s testimony, the materials I cite this report, and my own specialized knowledge and experience, it is my opinion that OEMs would not have had significant or lasting concerns in 2007 (or later years) if the 37 Java API packages had been licensed under the GPL-2.0-CE license.

49. I also disagree with Dr. Kemerer’s statement in paragraph 182 that “[a]s of December 2015 Google had not used GPL-licensed code in any prior version of Android.” To the contrary, as discussed above, Android incorporates a Linux kernel that is licensed under the GPL-2.0 license. And while Dr. Kemerer purports to rely on Mr. Ghuloum’s testimony in support of his (incorrect) conclusion, Mr. Ghuloum in fact testified that Android *did* use GPL-licensed code.<sup>22</sup>

**D. Dr. Kemerer conflates the GPL-2.0-CE license with the LGPL license.**

50. In paragraphs 169-173, Dr. Kemerer similarly errs in citing statements posted on an Android website regarding the LGPL license as support for his conclusions regarding Google’s willingness to include software licensed under the GPL-2.0-CE license.

51. I discussed these statements in my opening report at paragraph 48, where I explained: In response to inquiries regarding the choice of the Apache-2.0 license as the preferred Android license, Google published the statement copied below. As explained in the statement, Google selected the permissive Apache-2.0 license over a version of the weak-copyleft LGPL license to enable broader, less-encumbered use of the Android software on mobile devices. Google identified two specific obligations in the LGPL license that cause challenges or concerns for commercial software and commercial device manufacturers: (a) the

---

<sup>22</sup> Ghuloum depo. at 34:20-23, 62:8-10.

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

obligation to enable downstream recipients of software linked with an LGPL library to replace that library (standalone or as embedded on device) with their own preferred version; and (b) the obligation to permit downstream modification and reverse engineering of LGPL libraries.

We are sometimes asked why Apache Software License 2.0 is the preferred license for Android. For userspace (that is, non-kernel) software, we do in fact prefer ASL2.0 (and similar licenses like BSD, MIT, etc.) over other licenses such as LGPL.

Android is about freedom and choice. The purpose of Android is promote openness in the mobile world, and we don't believe it's possible to predict or dictate all the uses to which people will want to put our software. So, while we encourage everyone to make devices that are open and modifiable, we don't believe it is our place to force them to do so. Using LGPL libraries would often force them to do just that.

Here are some of our specific concerns:

- LGPL (in simplified terms) requires either: shipping of source to the application; a written offer for source; or linking the LGPL-ed library dynamically and allowing users to manually upgrade or replace the library. Since Android software is typically shipped in the form of a static system image, complying with these requirements ends up restricting OEMs' designs. (For instance, it's difficult for a user to replace a library on read-only flash storage.)
- LGPL requires allowance of customer modification and reverse engineering for debugging those modifications. Most device makers do not want to have to be bound by these terms. So to minimize the burden on these companies, we minimize usage of LGPL software in userspace.
- Historically, LGPL libraries have been the source of a large number of compliance problems for downstream device makers and application developers. Educating engineers on these issues is difficult and slow-going, unfortunately. It's critical to Android's success that it be as easy as possible for device makers to comply with the licenses. Given the difficulties with complying with LGPL in the past, it is most prudent to simply not use LGPL libraries if we can avoid it.

The issues discussed above are our reasons for preferring ASL2.0 for our own code. They aren't criticisms of LGPL or other licenses. We are passionate about this topic, even to the point where we've gone out of our way to make sure as much code as possible is ASL2.0 licensed. However, we love all free and open source licenses, and respect others' opinions and



**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

preferences. We've simply decided ASL2.0 is the right license for our goals.<sup>23</sup>

52. As discussed in paragraph 49 of my opening report, linking (dynamically or statically) with software licensed under the GPL-2.0-CE license does not raise the same challenges and concerns that Google raised with respect to the LGPL license.

53. As I explained in paragraph 56 of my opening report, the LGPL license includes certain compliance obligations that may be inconvenient to closed-source product distributors and may present challenges and distribution deterrents for manufacturers and distributors of devices (such as televisions, mobile phones, and kitchen appliances) including embedded LGPL-licensed software.

54. By way of contrast, the GPL-2.0-CE license excuses distributors of software merely linking to GPL-2.0-CE licensed software from not just the GPL-2.0 license’s copyleft effects but from all compliance requirements relating to the GPL-2.0-CE licensed software. Specifically, for independent modules that link to the GPL-2.0-CE licensed code, the requirement is merely that, for each independent module, you comply with the “terms and conditions of the license of that module.”

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of

---

<sup>23</sup> <https://source.android.com/source/licenses.html>

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.<sup>24</sup>

55. In fact, none of the three bullet-pointed concerns that Google raised with using the LGPL license for the Android platform apply to code licensed under the GPL-2.0-CE license.

56. First, Google expressed concern about the requirement that, if source code for the linking application is not made available, then the LGPL-licensed library must be linked to “dynamically” to enable recipients to “manually upgrade or replace the library.” However, the GPL-2.0-CE license does not impose such an obligation and excuses software that is “linking” to the GPL-2.0-CE code from any requirements, including a requirement to allow recipients to manually upgrade or replace the LGPL-licensed library.

57. Second, Google expressed concern that the LGPL license requires “allowance of customer modification and reverse engineering for debugging those modifications.” The GPL-2.0-CE license places no such requirement on software qualifying for the Classpath Exception.

58. Third, Google noted that “LGPL libraries have been the source of a large number of compliance problems for downstream device makers and application developers.” But because the GPL-2.0-CE license eliminates all compliance requirements for independent modules that link with the GPL-2.0-CE software, GPL-2.0-CE libraries do not cause such compliance problems.

59. In sum, none of the concerns that Google’s webpage raises with respect to the *LGPL* license apply to software linked to software licensed under the *GPL-2.0-CE* license. In my opinion, Dr. Kemerer erred by relying on Google’s comments about the LGPL license to conclude that Google would not have been willing to license the 37 Java API packages in OpenJDK under the GPL-2.0-CE license.

---

<sup>24</sup> <https://www.gnu.org/software/classpath/license.html>

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

60. Similarly, in paragraph 173 of his report, Dr. Kemerer refers to a slide in a Google presentation that states, “we want to keep GPL out of user-space.”<sup>25</sup> That bullet point is on a slide that explains why Google built a custom “libc” library. The most commonly used libc library is the GNU C Library (sometimes referred to as glibc), which is licensed under the LGPL license. Accordingly, the slide’s mention of “GPL” refers to the LGPL license rather than the GPL-2.0-CE license. As noted explained above, linking to software licensed under the GPL-2.0-CE license does not raise the same compliance concerns as linking to software licensed under the LGPL license.

61. For these reasons, it is my opinion that the conclusions Dr. Kemerer makes based on evidence about the LGPL license are incorrect as applied to the GPL-2.0-CE license. Linking with libraries available under the GPL-2.0-CE license raises none of the three concerns that Google highlighted with respect to the LGPL license. The LGPL license examples cited by Dr. Kemerer therefore do not change my opinion, based upon Anwar Ghuloum’s testimony, the materials I cite this report, and my own specialized knowledge and experience, that OEMs would not have had significant or lasting concerns in 2007 (or later years) with using the 37 Java API packages if they had been licensed under the GPL-2.0-CE license.

---

<sup>25</sup> GOOGLE-22-00280894.

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

**V. CONCLUSION**

62. For the reasons explained above, I disagree with Dr. Kemerer’s conclusion that Google and/or its partners would not have accepted licensed alternatives to the 37 Java API packages in the 2007-2010 timeframe. To the contrary, it is my opinion that Google and its partners would not have had significant or lasting concerns in 2007 (or later years) if the 37 Java API packages in Android had been licensed under the GPL-2.0-CE license.

Executed on the 8th of February, 2016 in Novato, CA.

A handwritten signature in black ink, appearing to read 'Andrew Hall', is written over a horizontal line.

Andrew Hall

**HIGHLY CONFIDENTIAL – ATTORNEYS’ EYES ONLY**

**APPENDIX D – DOCUMENTS EXAMINED DURING REPORT PREPARATION**

I examined the following documents, published materials, and websites in the course of preparing this report include:

- Documents cited in Appendix B to my opening report
- Redacted copy of Dr. Kemerer’s January 8, 2016 expert report
- Documents cited in this rebuttal report and my opening report